2016

Cyberbullying Detection with a Pronunciation Based Convolutional Neural Network

Xiang Zhang
Clemson University

Jonathan Tong
Clemson University

Nishant Vishwamitra
Clemson University

Elizabeth Whittaker
Clemson University

Joseph P. Mazer
Clemson University

See next page for additional authors

Authors

Xiang Zhang, Jonathan Tong, Nishant Vishwamitra, Elizabeth Whittaker, Joseph P. Mazer, Robin Kowalski, Hongxin Hu, Feng Luo, Jamie C. Macbeth, and Edward Dillon

2016 15th IEEE International Conference on Machine Learning and Applications

Cyberbullying Detection with a Pronunciation Based Convolutional Neural Network

Xiang Zhang[1], Jonathan Tong[1, *], Nishant Vishwamitra[1], Elizabeth Whittaker[3],
Joseph P. Mazer[2], Robin Kowalski[3], Hongxin Hu[1], Feng Luo[1]
[1]School of Computing, [2]Department of Communication Study, [3]Department of Psychology
Clemson University
luofeng@clemson.edu

Jamie Macbeth
School of Engineering
Fairfield University

Edward Dillon
Dept. of Computer and Information Sciences and Eng.
University of Florida

***Abstract*—Cyberbullying can have a deep and long lasting impact on its victims, who are often adolescents. Accurately detecting cyberbullying helps prevent it. However, the noise and errors in social media posts and messages make detecting cyberbullying very challenging. In this paper, we propose a novel pronunciation based convolutional neural network (PCNN) to address this challenge. Upon observing that the pronunciation of misspelled words in informal online conversations is often unchanged, we used the phoneme codes of the text as the features for a convolutional neural network. This procedure corrects spelling errors that did not alter the pronunciation, thereby alleviating the problem of noise and bullying data sparsity. To overcome class imbalance, a common problem in cyberbullying datasets, we implement three techniques that include threshold-moving, cost function adjusting, and a hybrid solution in our model. We evaluate the performance of our models using two cyberbullying datasets collected from Twitter and Formspring.me. The results of our experiment show that PCNN can achieve improved recall and precision compared to baseline convolutional neural networks.**

I. INTRODUCTION

The rise of social media has significantly influenced our lives. However, this puts adolescents at risk to becoming victims of online misconduct, especially cyberbullying. Cyberbullying refers to an aggressive, intentional act conducted by either a group or an individual in cyberspace using information and communication technologies (e.g. e-mail, mobile phone, and social networks) repeatedly or over time against victims who cannot easily defend themselves [1]. According to a 2015 report from the Cyberbullying Research Center, about one-third of the high school students from random samples have experienced cyberbullying [2].

Unlike in traditional bullying, techniques and forms used by cyberbullies change rapidly and is more harmful and harder to detect [3]. For example, it is easy to anonymously spread rumors about people online, and there is a low risk of being caught. Thus, it is necessary to detect cyberbullying in order to protect adolescents.

Unlike video and image based methods, text based cyberbullying is the most common form used by perpetrators. Moreover, other forms are usually combined with bullying text. Thus, we focus on detecting textual cyberbullying in this study. Textual cyberbullying detection methods can be divided into two categories: keywords based and artificial intelligence (AI) based [4].

The simplest way is the keyword method, which uses keywords to search for sensitive content within a text. Although the idea is straightforward, this method can still obtain a high precision score by using informative query terms and leveraging the internet searching [5, 6]. However, keywords themselves are far from representative of all cyberbullying content. Thus, keywords based approaches have difficulty achieving high recall, a more important metric than precision and accuracy for cyberbullying detection. This is because it is better to detect more cyberbullying posts, even if there are false positives, than to have a high precision but only find a fraction of the cyberbullying posts [7]. In addition, accuracy is not a useful metric in this context because the classifier can easily achieve a relatively high accuracy of 93% by predicting all the samples as negatives (non-bullying) but have zero recall.

The AI method is more complex. The three core components of AI, representation, inference, and learning, create three corresponding research directions for cyberbullying detection methods [4]. To be specific, most methods are based on supervised machine learning classifiers [7-20]. Our cyberbullying detection methods also belong to the machine learning based approach.

TABLE I. EXAMPLES OF NOISY DATA

Examples	**Noise Type**
“! w@nN@ l!qqhH+ y0 d!(K 0N f!r3 N d3nN sM0k3 *!t w!+ m@ v@q!n@”*	Symbols
“wHy yUhH w0N+ fU(K m3 !N d@ @$$ h0l3 ???”	Symbols
“lol yew on sum otha shxt nd not even dressed in all black”	Intended typos
“im sur3 sh3 d0nt want y0u”	Numbers
“iloveyourpenis”	Concatenation

[a] The examples are from Formspring.me website.

The characteristics of posts and messages with bullying content make the detection of cyberbullying very challenging. First, as shown in Tab. I, these texts have many words with

[*] Intern from D.W. Daniel High School, Central, SC USA

978-1-5090-6167-9/16 $31.00 © 2016 IEEE
DOI 10.1109/ICMLA.2016.71

incorrect spellings and symbol. However, we observed that those misspelled words are often informative. In addition, many sentences made up of symbols contain bullying.

Second, the distribution of the classes within the data is imbalanced and the proportion of bullying content varies from different websites. For example, approximately 17% of the messages in the samples from Formspring.me, a question and answer based social network, contain bullying content [7]. The ratio dropped to approximately 5% after we parsed the messages into individual sentences on our analysis of the same data.

The motivation of our work is the need for a *practical*, *robust*, and *universal* cyberbullying detection classifier with *high performance*. In this study, we propose to use the pronunciation of words within the texts as the features for a convolutional neural network (CNN), a classifier that has shown high performance on natural language processing, e.g. sentiment analysis [21]. The pronunciation conversion corrects spelling errors that did not change the original pronunciation of the word by mapping each word to phonetic code, which also reduces the size of the feature space. Our new pronunciation based convolutional neural network (PCNN) can alleviate the noise in social media text and improve classification performance. In order to overcome the class imbalance problem, we adopted three techniques: threshold-moving (TM) and cost function adjusting (CFA), and a hybrid solution (TM CFA) [22]. We tested our model on the Twitter dataset used in [18] and the Formspring dataset used in [7] to have a clear comparison between our approach and other works.

II. Related Works

Kontostathis et al. [6] analyzed cyberbullying corpuses using the bag-of-words model to find the most common used terms by cyberbullies and used them to create queries capable of reaching a precision of 91.25% on average. Lempa et al. [5] developed an Android application, embedded with two methods, to implement the cyberbullying detection. One method is built on a brute force search algorithm search for sensitive words and phrases within the text. The other method extracts words and phrases as seed words and detects cyberbullying online with keyword categorization and relevance matching. The top precision of both methods reaches 89% and 91%, respectively [5].

Regarding classifier design, researchers have tested various classifiers, including Naïve Bayes, C4.5 decision tree, Random forests, and SVM with different kernels on corpuses collected from popular social networks, such as Twitter and YouTube [7, 11]. Reynolds et al. [7] found that both the C4.5 decision tree and 3-nearest neighbor classifiers can reach a recall of 78.5% on the text-based dataset collected from Formspring.me, a question and answer based social network. Bullying posts (positives) were duplicated 10 times to compensate for the imbalance within the data. However, this oversampling method is unreliable since it exaggerates the occurrence rate of the positive samples [7]. On the Twitter dataset, Xu et al. [11] showed that SVM with a linear kernel using unigrams and bigrams as features can achieve a recall of 79% and a precision of 76%. Other works are focused on ensemble methods such as cooperative and hybrid classifiers [12, 15, 19]. Dadvar et al. [12, 15] introduced two approaches to combine machine learning methods and expert systems. The different combinations depend on which classifier's output is used as the input of the other. An accuracy metric called the area under the curve (AUC) was used to evaluate their approach. The hybrid system made up of expert system and Naïve Bayes classifier, achieving their highest AUC score of 0.76. Mangaonkar et al. [19] evaluated 15 cooperative classifier combinations, including heterogeneous, homogeneous, and selective cooperation with different parallelisms. These ensemble classifiers are extremely complex and tuning the hyperparameters is difficult.

For feature selection, various textual content based features, such as the basic bag-of-words and advanced sentiment prediction, were used as the input to classifiers [14, 17, 18]. Kasture took advantage of a psychometric feature analysis tool called Linguistic Inquiry and Word Count (LIWC) used it as a feature extractor. These features were used to train a variety of classifiers and the best performance reached 96.3% recall and 98.4% precision on Random Forests using 10-fold cross validation on the Twitter dataset [18]. To detect the cyberbullying and cyberstalking in emails and messages, Ghasem et al. [17] selected the 500 most informative words as the feature vector and achieved an F1 score of approximately 95% on SVM and a neural network classifier. Nahar et al. [14] introduced a weighted TFIDF feature extractor and used LIBSVM with a linear kernel to detect cyberbullying content in three social networks: Kongregate, Slashdot, and Myspace. The experiment results show that their feature design significantly improved the performance of the baseline LIBSVM. For example, the recall jumped from 25% to 98% on the Myspace dataset. However, oversampling was used to handle the imbalance problem, which is not a useful method in real-world implementations [14].

To further improve performance, many researchers implemented context-based features such as user profile information and online duration [8-10, 12, 13, 16, 20]. Dadvar et al. [9, 10, 12, 13] investigated incorporating user information as features to improve the performance. They established a comprehensive, context-based feature set covering age, behavior, cross-platform information, and activity history. These features were first tested by SVM and then used to build a hybrid detection approach including an expert system as mentioned above. Patterns of social network structures involving user behavior were used to detect and analyze cyberbullying [8, 16, 20]. Features like the number of friends, relation centrality, and bullying propagation were investigated and used to aid the detection. Their research results show that human relationships and action dynamics within social network structures can be taken into account to improve the results of cyberbullying detection and prediction. However, this type of context information is usually unavailable due to privacy protections. Thus, an effective and robust cyberbullying detection method should be able to perform well without this information.

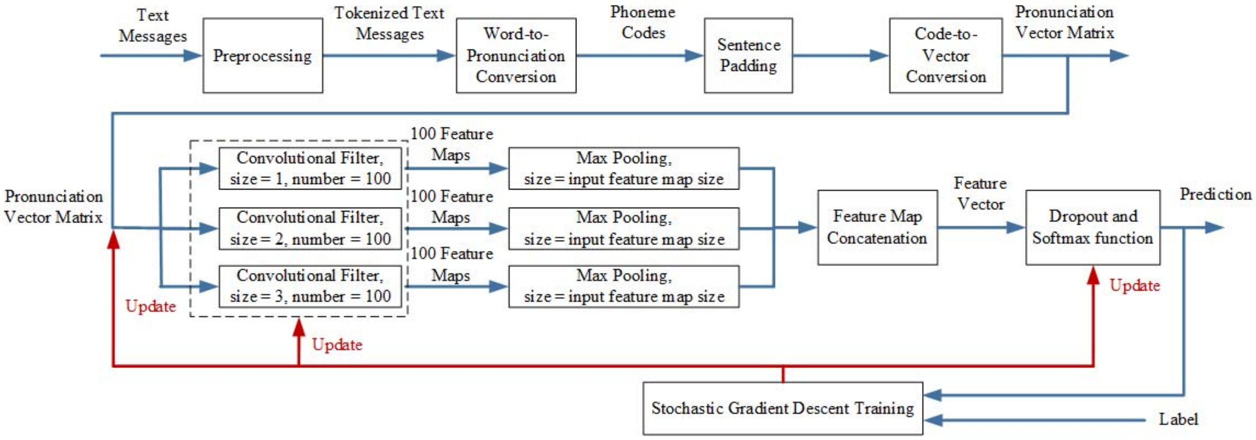
Text Messages
Preprocessing
Tokenized Text Messages
Word-to-Pronunciation Conversion
Phoneme Codes
Sentence Padding
Code-to-Vector Conversion
Pronunciation Vector Matrix
Pronunciation Vector Matrix
Convolutional Filter, size = 1, number = 100
Convolutional Filter, size = 2, number = 100
Convolutional Filter, size = 3, number = 100
100 Feature Maps
100 Feature Maps
100 Feature Maps
Max Pooling, size = input feature map size
Max Pooling, size = input feature map size
Max Pooling, size = input feature map size
Feature Map Concatenation
Feature Vector
Dropout and Softmax function
Prediction
Update
Update
Update
Stochastic Gradient Descent Training
Label

Fig. 1. The architecture of PCNN

III. Method Description

A. Data Collection

Due to the relatively limited amount of research in cyberbullying detection, there are no well-established benchmark datasets to test various approaches. The datasets used in previous publications differ significantly in size, source, and format. We decided to use the two datasets used in [18] and [7] for the following reasons.

The Twitter dataset was used since Twitter is a popular platform and the dataset has been recently created and analyzed [18]. However, the dataset contains only 1313 messages, and the bullying content proportion, approximately 38.8%, is significantly higher than it would be under realistic conditions.

Another dataset, collected from the social network Formspring.me and used in [7], was chosen to give an additional evaluation our approach. 13,000 messages were collected and then labeled by a web service called Amazon Mechanical Turk, where three workers each voted on whether or not a document contains bullying content. Thus, every message has a corresponding number of votes from the workers. Approximately 6.6% of the messages were labeled as bullying posts by at least two workers. We parsed the messages from the original dataset into sentences and relabeled the messages containing at least one vote. This resulted in 23,243 sentences in which 1,623, or approximately 7%, are labeled as bullying messages.

B. Data Preprocessing and Word-to-Pronunciation Conversion

The Twitter dataset had already been preprocessed by removing usernames, hashtags, and hyperlinks from the Tweets [18]. The Tweets were then converted into plain text by replacing accented characters and removing non-alphanumeric tokens, excluding the apostrophe. After preprocessing, the maximum length of any Tweet was 33 words, and only approximately 15% of the words of the dataset cannot be found in a dictionary.

We performed similar preprocessing on the Formspring dataset. Specifically, we removed irrelevant words like hyperlinks, user indicators (“Q:” and “A:”) and non-alphanumeric tokens. Then, a term-compression operation was performed to ensure that there are no more than 2 consecutive occurrences of any character in a word. For example, “coolll,” “bitchhh,” and “guesss” become “cooll,” “bitchh,” and “guess”. This simple technique helps the pronunciation conversion procedure to group misspelled words with the same meaning and pronunciation together with the corrected word.

After preprocessing the datasets, we created the phonetic representation of each word using eSpeak, an open source speech synthesizer software [23]. This conversion was based on pronunciation rules and a dictionary lookup list, both of which can be manually modified to better suit the purpose and research context.

The phoneme strings can be encoded using ASCII code or the International Phonetic Alphabet (IPA), which uses characters from the Latin alphabet. Here, we used ASCII code to remain consistent with the original plain text. Some examples of the conversion are on Tab. II.

TABLE II. Examples of the Word-to-Pronunciation Conversion

Word and phrases	Phonetic code	Effect on the data
“fuck, fuc, fuk”	*f'Vk*	Positive
“fuckk”	*f'Vkk*	Neutral
“shitt, shit”	*S'It*	Positive
“suck, suk, suc”	*s'Vk*	Positive
“dik, dic, dick”	*d'Ik*	Positive
“guesss, guess”	*g'Es*	Positive
“bitchh, bitch, bich”	*b'ItS*	Positive
“cool, cooll”	*k'u:l*	Positive
“cum, come”	*k'Vm*	Negative

b. The examples are from Twitter and Formspring datasets.

The word-to-pronunciation conversion can map some misspelled words to the pronunciation of the corrected word. We observed that, especially in bullying posts, perpetrators tended to use slang or intentionally misspell words when insulting others. For example, five out of six words in the phrase "did u now tht ur ugli" are typos. However, the pronunciation of these misspelled insults usually remains unchanged. This means that the word-to-pronunciation conversion can generate the same phonetic code for "ugli" and "ugly," effectively "correcting" the misspelled word.

On the other hand, this conversion can create noise by mapping a bad word such as "cum" and a normal word such as "come" to the same phonetic representation. However, we believe the benefits of this procedure outweigh the costs since correctly maps words much more often than it creates noise. Furthermore, surrounding context words could be used to distinguish words like "cum" and "come."

After preprocessing the data and applying the word-to-pronunciation conversion, the phonetic representation of each word was converted to a randomly initialized 300-dimensional vector. Then, a zero vector with the same dimension was used to pad each the sentences so that they are all the same length. Finally, each sentence was projected to a matrix with the same size.

C. CNN and PCNN

Convolutional neural networks (CNN), originally created for image processing, have performed very well in natural language processing (NLP), especially in sentiment analysis and question classification [21, 24, 25]. Inspired by their powerful feature representation capability, flexible structure, and high efficiency for training using a GPU, we adopted CNN as the baseline classifier. To have a clear performance comparison between PCNN and the baseline CNN, we used the same model architecture in [21]. As shown in the PCNN architecture diagram, only one layer of convolution and max-pooling was used with three different filter sizes. The sizes of the three convolutional filters were chosen to be 1, 2, and 3, slightly differing from the filters in [21]. The filter sizes were chosen based on how many consecutive words were necessary to detect bullying content. The convolutional operation on m consecutive words is given in (1).

$$h_i = f(\boldsymbol{w}_c \boldsymbol{x}_{i:i+m} + b_c) \tag{1}$$

Here, $\boldsymbol{x}_{i:i+m}$, h_i, $\boldsymbol{w}_c$, b_c, and f are the embedding matrix of m words, the feature value generated by the operation, the weight and bias of the corresponding convolutional filter, and the activation function, respectively. We used the linear rectifier unit as the activation function.

A max-pooling operation was applied to all the features from one convolutional filter. Then, the features were concatenated into $\boldsymbol{h}$, a feature vector with dimensions equal to the number of filters applied. A softmax layer with dropout was applied to the output of the pooling layer to predict the class probability, P, as follows:

$$P\,(Y{=}i \mid X, \boldsymbol{\theta}) = softmax_i\,(\boldsymbol{w}_s \boldsymbol{h} + b_s) \tag{2}$$

Here X, $\boldsymbol{h}$, Y, $\boldsymbol{w}_{\mathrm{s}}$, b_{s}, i, and $\boldsymbol{\theta}$ are the input embedding matrix, feature vector from the convolutional and pooling layer, class prediction, weights of the penultimate layer, corresponding bias, class number, and parameter set, respectively.

We used two separate CNN to establish a baseline. Three hundred dimensional word-embedding based on Google's word-to-vector was used to create the feature set for the first baseline CNN, which we named CNN Pre-trained. Randomly generated vectors were used to create the feature set for the second baseline CNN, which we named CNN Random. For PCNN, the phoneme codes were randomly initialized into vectors for the feature set. All the embedding for CNN and PCNN was updated during the training process based on the stochastic gradient descent [26].

Our method and the structure of PCNN are shown in Fig. 1.

D. Techniques for Handling Class Imbalance

Unlike the movie reviews used in sentiment analysis, the class distribution is imbalanced for most cyberbullying related datasets. For example, only about 6.6% messages in Formspring dataset were labeled as bullying by two voters. The class imbalance within the dataset creates two problems.

First, the small percentage of positive samples makes it difficult to detect them, especially when the dataset is small. Furthermore, the lack of sufficient samples for unique instances of bullying makes it nearly impossible for classifiers to recognize them.

Second, the most commonly used cost functions of CNN, the one we used being the negative log likelihood (NLL), were designed to only improve the accuracy rather than recall or precision. Consequently, CNN models have a bias towards the dominating class.

There are three methods for dealing with the class imbalance problem: oversampling or undersampling the dataset, modifying the classifier to be cost sensitive, and training on only one class [22, 27]. However, the sampling technique will change the proportion of the classes, causing the data to no longer represent realistic conditions, and the one class classification method is only useful for detecting anomalies and outliers in the dataset.

Since these two methods are not useful for most datasets, we chose to modify the classifier to be cost sensitive. There are three ways to implement this: threshold-moving, cost function adjusting, and a hybrid solution. Threshold-moving replaces Bayes estimation with maximum likelihood estimation in order to compensate for the great difference between the prior probabilities of the two classes. The prediction using this method is calculated by dividing the prediction of the classes by their prior probabilities, $P(Y{=}i)$. It is implemented as follows:

$$Y_{predict} = argmax_i\,(P(Y = i \mid X, \boldsymbol{\theta}) \,/\, \mathrm{P}(Y = i)) \tag{3}$$

Re-normalization could be added, but it does not affect the prediction result.

Cost function adjusting aims to modify the cost function of for the stochastic gradient descent training so that each element of the minority class can cause more parameter optimization than each element of the majority class does to compensate for the class imbalance. The new cost function is given as follows:

$$cost = -(1 / Kn) \cdot \Sigma^{n}_{i=1}(log\ (P(Y=i \mid X, \boldsymbol{\theta})) / P(Y = i)) \qquad (4)$$

$P\ (Y=i \mid X, \boldsymbol{\theta})$ is the predicted probability of class i when the input embedding matrix is X and the model parameter set is $\boldsymbol{\theta}$. $P(Y=i)$ is the prior probability of class i, n is the size of the mini-batch, and K is the number of classes.

IV. Experiment Result

We used recall, precision, F1 score and accuracy as metrics to evaluate the performance of our models. All of these metrics are based on the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The formulas for calculating the metrics are as follows:

$$Recall = TP / (TP + FN) \qquad (5)$$

$$Precision = TP / (TP + FP) \qquad (6)$$

$$F1 = 2 * Recall * Precision / (Recall + Precision) \qquad (7)$$

We used 5-fold cross-validation for the Formspring dataset. However, we used ten-fold cross-validation on the Twitter data in order to remain consistent with [18]. Two baseline CNN classifiers and PCNN were tested and compared with the other classifiers used in [18]. The code was written in Python and the neural network components were based on the Theano package and Kim's work [21, 28].

A. Comparison of Classification Performance

Tab. III shows the results of our methods on the Twitter dataset compared to previous work based on LIWC features in [18]. It shows that PCNN outperforms all models on all metrics in the original paper and is slightly better than CNN with randomly generated word embedding. In addition, PCNN and CNN Random performed better than CNN with pre-trained Google word vectors. This may be because the corpus used for pre-training was not specific to cyberbullying detection.

TABLE III. Comparison of Methods Used on The Twitter Dataset

Model	Precision	Recall	Accuracy	F1 score
Random forests	0.984	0.963	-	0.973
SVM	0.986	0.912	-	0.948
Multilayer Perceptron	0.951	0.939	-	0.945
J 48 Decision Tree	0.947	0.941	-	0.944
CNN Pre-trained	0.973	0.937	0.974	0.955
CNN Random	**0.994**	0.962	0.988	0.978
PCNN	0.991	**0.970**	**0.989**	**0.980**

[c] Results are average of 10-fold cross validation

[d] The accuracy of first four classifiers from [18] is not given.

TABLE IV. Comparison of Methods Used on the Formspring Data

Model	Precision	Recall	Accuracy	F1 score
CNN Pre-trained	0.728	0.364	0.964	0.485
CNN Random	0.728	0.429	0.966	0.540
PCNN	**0.740**	**0.453**	**0.968**	**0.562**

Results are average of 5-fold cross-validation

Tab. IV summarizes results of the three approaches evaluated on the Formspring dataset. PCNN outperformed the two baseline CNN models in all metrics, demonstrating the benefit of using the word to pronunciation conversion. Despite the excellent results on the Twitter dataset, the overall classification performance on the Formspring data was much lower. This may be due to the severe noise and class imbalance in the Formspring dataset. For example, approximately 55% of the words in the Formspring dataset vocabulary cannot be found in the dictionary while only 15% of the words in the Twitter dataset are misspelled.

B. Comparison of Techniques for Handling Class Imbalance

Tab. V shows the results of the three CNN models on the Twitter dataset using the different class imbalance handling techniques: threshold-moving, cost function adjusting, and a hybrid solution. These techniques slightly improved recall, and the combination of TM and CFA performed the best out of the three. Furthermore, TM CFA PCNN can improve the overall performance and outperforms the two baseline CNN models. However, the improvement is insignificant since the degree of class imbalance in the Twitter dataset is low and the recall is already very high. Thus, these techniques need to be evaluated on a noisier and more imbalanced dataset.

Tab. VI gives the corresponding results on the Formspring dataset. It shows that all three techniques enhanced the recall at the cost of precision and even accuracy. Among them, CFA improved the overall classification performance the most, increasing recall and F1 score without hurting accuracy. Moreover, PCNN obtained the highest recall and F1 score than others when using cost function adjusting.

TABLE V. Handling Class Imbalance on the Twitter Dataset

Technique	Model	Precision	Recall	Accuracy	F1 score
TM	CNN Pre-trained	0.910	0.943	0.956	0.926
	CNN Random	0.984	0.961	0.985	0.972
	PCNN	**0.989**	**0.972**	**0.989**	**0.980**
CFA	CNN Pre-trained	0.954	0.946	0.971	0.950
	CNN Random	**0.992**	0.960	0.986	0.976
	CFA PCNN	0.991	**0.972**	**0.990**	**0.981**
TM CFA	CNN Pre-trained	0.919	0.949	0.960	0.934
	CNN Random	0.986	0.965	0.986	0.975
	PCNN	**0.991**	**0.975**	**0.990**	**0.983**

TABLE VI. HANDLING CLASS IMBALANCE ON THE FORMSPRING DATASET

Technique	Model	Precision	Recall	Accuracy	F1 score
TM	CNN Pre-trained	**0.328**	0.602	0.923	0.425
	CNN Random	0.280	0.694	0.894	0.399
	PCNN	0.305	**0.717**	**0.902**	**0.428**
CFA	CNN Pre-trained	0.440	0.529	0.947	0.480
	CNN Random	**0.562**	0.558	**0.960**	0.560
	PCNN	0.540	**0.606**	0.958	**0.571**
TM CFA	CNN Pre-trained	0.168	0.733	0.818	0.273
	CNN Random	0.203	0.778	0.846	0.322
	PCNN	**0.254**	**0.787**	**0.881**	**0.384**

The results on both datasets show that cost function adjusting is an effective technique to handle datasets with class imbalance. In addition, the word-to-pronunciation conversion contributes to the increase the recall without other loss.

V. CONCLUSION AND FUTURE WORK

We have proposed a novel pronunciation based convolutional neural network for detecting cyberbullying. We have compared our approach with two baseline CNN models and other classifiers using two datasets, each with different degrees of noise and class imbalance. Our approach showed high performance on the given datasets. Furthermore, three techniques for overcoming class imbalance have been implemented and evaluated. The results show that the PCNN with cost function adjusting is a very effective solution.

In the future, we plan to enhance the effectiveness of the word-pronunciation conversion and connect pronunciation features with the CNN model more tightly.

ACKNOWLEDGMENT

This work was partially supported by grants from National Science Foundation (NSF-CNS-1537924 and NSF-IIS-1527421).

REFERENCES

[1] D. L. Espelage and S. M. Swearer, "Research on school bullying and victimization: What have we learned and where do we go from here?," *School psychology review,* vol. 32, pp. 365-384, 2003.

[2] J. W. Patchin. (2015). *2015 Cyberbullying Data*. Available: http://cyberbullying.org/2015-data

[3] J. W. Patchin and S. Hinduja, "Measuring cyberbullying: Implications for research," *Aggression and Violent Behavior,* vol. 23, pp. 69-74, 2015.

[4] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of Textual Cyberbullying," *The Social Mobile Web,* vol. 11, p. 02, 2011.

[5] P. Lempa, M. Ptaszynski, and F. Masui, "Cyberbullying Blocker Application for Android," presented at the 7th Language & Technology Conference (LTC'15), Poznan, Poland, 2015.

[6] A. Kontostathis, K. Reynolds, A. Garron, and L. Edwards, "Detecting cyberbullying: query terms and techniques," in *Proceedings of the 5th annual acm web science conference*, 2013, pp. 195-204.

[7] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," in *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, 2011, pp. 241-244.

[8] M. Honjo, T. Hasegawa, T. Hasegawa, K. Mishima, T. Suda, and T. Yoshida, "A framework to identify relationships among students in school bullying using digital communication media," in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, 2011, pp. 1474-1479.

[9] M. Dadvar and F. de Jong, "Cyberbullying detection: a step toward a safer Internet yard," in *Proceedings of the 21st International Conference on World Wide Web*, 2012, pp. 121-126.

[10] M. Dadvar and F. Jong, "Improved Cyberbullying Detection Through Personal Profiles," presented at the International Conference on Cyberbullying, 2012.

[11] J.-M. Xu, K.-S. Jun, X. Zhu, and A. Bellmore, "Learning from bullying traces in social media," in *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 2012, pp. 656-666.

[12] M. Dadvar, D. Trieschnigg, and F. Jong, "Expert knowledge for automatic detection of bullies in social networks," presented at the Proceedings of the 25th Benelux Conference on Artificial Intelligence, Delft, the Netherlands, 2013.

[13] M. Dadvar, D. Trieschnigg, R. Ordelman, and F. de Jong, "Improving cyberbullying detection with user context," in *European Conference on Information Retrieval*, 2013, pp. 693-696.

[14] V. Nahar, X. Li, and C. Pang, "An effective approach for cyberbullying detection," *Communications in Information Science and Management Engineering,* vol. 3, p. 238, 2013.

[15] M. Dadvar, D. Trieschnigg, and F. de Jong, "Experts and machines against bullies: A hybrid approach to detect cyberbullies," in *Canadian Conference on Artificial Intelligence*, 2014, pp. 275-281.

[16] Q. Huang, V. K. Singh, and P. K. Atrey, "Cyber bullying detection using social and textual analysis," in *Proceedings of the 3rd International Workshop on Socially-Aware Multimedia*, 2014, pp. 3-6.

[17] Z. Ghasem, I. Frommholz, and C. Maple, "Machine learning solutions for controlling cyberbullying and cyberstalking," *J Inf Secur Res,* vol. 6, pp. 55-64, 2015.

[18] A. S. Kasture, "A predictive model to detect online cyberbullying," Auckland University of Technology, 2015.

[19] A. Mangaonkar, A. Hayrapetian, and R. Raje, "Collaborative detection of cyberbullying behavior in Twitter data," in *2015 IEEE International Conference on Electro/Information Technology (EIT)*, 2015, pp. 611-616.

[20] A. Squicciarini, S. Rajtmajer, Y. Liu, and C. Griffin, "Identification and characterization of cyberbullying dynamics in an online social network," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, 2015, pp. 280-285.

[21] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882,* 2014.

[22] A. Dalyac, M. Shanahan, and J. Kelly, "Tackling class imbalance with deep convolutional neural networks," *Dept. Comput. Sci., Imperial College London, London, UK, Tech. Rep,* 2014.

[23] *eSpeak*. Available: http://espeak.sourceforge.net/

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, pp. 2278-2324, 1998.

[25] W.-t. Yih, X. He, and C. Meek, "Semantic Parsing for Single-Relation Question Answering," in *ACL (2)*, 2014, pp. 643-648.

[26] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701,* 2012.

[27] D. Masko and P. Hensman, "The impact of imbalanced training data for convolutional neural networks," School of Computer Science and Communication, KTH ROYAL INSTITUTE OF TECHNOLOGY, STOCKHOLM, SWEDEN, 2015.

[28] T. T. D. Team, R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau*, et al.*, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv preprint arXiv:1605.02688,* 2016.